What we claim is:

1. A computer system comprising memory means, storage means, and an object-oriented software product, the software product containing an object-oriented extensible class hierarchy for the storage of transport phenomena simulation data, the class hierarchy comprising a first set of generic classes representing a plurality of object types and a second set of generic classes representing member variables for the object types, the extensible class hierarchy permitting the addition of additional object types and additional member variables without any modifications to the class hierarchy itself.

2. The computer system of claim 1 wherein the transport phenomena comprises one or more of momentum, energy, and mass transport within a subsurface hydrocarbon-bearing reservoir and between the subsurface hydrocarbon-bearing reservoir and one or more delivery locations at the earth's surface.

3. The computer system of claim 2 wherein the transport between a subsurface hydrocarbon-bearing reservoir and one or more of the delivery locations comprises one or more transport pathways, the transport pathways comprising at least one of production and injection well types and one or more facility types that are linked together to form a facility network through which hydrocarbon fluids are transported between the subsurface reservoir and the delivery locations.

4. The computer system of claim 3 wherein the facility types contained within the transport pathways comprise at least one facility selected from surface flowlines, manifolds, separators, valves, pumps, and compressors.

5. The computer system of claim 4 wherein a text file (Data Definitions File) contains the definitions of the possible facility types that can be included in a simulation model and the definitions of the possible member variable types for each facility type.

6. The computer system of claim 1 wherein the object-oriented software product comprises a graphical user interface whereby a user of the computer system defines a simulation model containing the specific network of wells and facility objects to simulate transport phenomena into and out of a specific hydrocarbon-bearing reservoir.

7. The computer system of claim 1 wherein additional data member types being defined by the user of the object-oriented software product computer system of claim 1 wherein a user of the computer system defines additional facility data members by a graphical user interface, said additional data members extending the functionality of the computer system in a user-customizable manner.

8. The object-oriented software product of claim 1 wherein the object-oriented software is written in C++.

9. The computer system of claim 1 wherein the storage means comprises an object-oriented database.

10. A method of simulating transport phenomena in a facility network using a computer system having memory means, storage means, and object-oriented software, the method comprising the steps of:
   (a) building a model comprising a facility network;
   (b) specifying values of the member variables for each facility; and
   (c) using the specified values of the member variables in a mathematical simulation of transport phenomena within the facility network as a function of time.

11. The method of claim 10 wherein the facility network is part of a larger simulation model, with said facility network being capable of exchanging fluids with at least one other part of the simulation model.

12. The method of claim 11 wherein the simulation model comprises a facility network and a hydrocarbon-bearing formation.

13. A method of simulating transport phenomena in a physical system comprising a hydrocarbon-bearing reservoir penetrated by a plurality of wells and surface facilities, connected to the wells, the method comprising:
    a. discretizing the physical system into a plurality of volumetric cells, wherein each volumetric cell is modeled as a node, and adjacent nodes being capable of exchanging fluid through connections between the nodes;
    b. using facility objects and member variable objects of the class hierarchy of claim 1 to model the nodes and connections in the portion of the discretized model that represents the wells and surface facilities;
    c. specifying the geometric and transport properties for each node and connection;
    d. specifying the initial conditions for each node and connection; and
    e. simulating as a function of time the transport phenomena in the discretized system.

14. An object-oriented software architecture having a plurality of classes that separates member variables from the classes to which the member variables logically belong, said software architecture comprising:
    a. a hierarchy of facility classes, wherein the most-specialized derived classes in the hierarchy of facility classes (Node, Connection, Compound, Well) are designed to generically represent the types of facilities that can be modeled;
    b. a hierarchy of member variable classes, wherein the most-specialized derived classes in the hierarchy of member variable classes (SystemAttributeValue, UserAttributeValue, RateConstraint, PressureConstraint) are designed to generically represent the types of member variables that a facility can have, said member variables being one of the following types: floating point scalar or array, an integer scalar or array, a string, a boolean, an enumerated type, a flow rate limit, and a pressure limit; and

c. a ValueUse class (206, Fig. 2) that has a many-to-one association with the base class in the hierarchy of data member classes (Value, 207, Fig. 2) and a many-to-one association with the base class in the hierarchy of facility classes (FacBase, 200, Fig. 2), such that each Value object has one or more references to the ValueUse objects that relate the usage of that Value object to the FacBase object that logically owns it and controls access to it.

15. The object-oriented software architecture of claim 14 wherein the complete hierarchy of classes includes classes in addition to the aforementioned facility classes and member variable classes, said additional classes provide the flexibility to reuse facility objects in multiple simulation realizations, with each distinct realization able to reuse all, some, or none of the member variable objects in one or more of the other realizations that use the facility object, said additional classes comprising:

a. a WmSystem controller class (217, Fig. 2) that has a direct one-to-many association to the base class in the hierarchy of facility classes (FacBase, 200, Fig. 2), such that an object of the WmSystem class in a collection of simulation realizations has direct references to the superset of all facility objects that are used in those simulation realizations;

b. a Case class (215, Fig. 2) that has a many-to-one association to the WmSystem class, such that an object of the WmSystem class can be used by many objects of the Case class, a Case object comprising the complete set of data for one simulation realization;

c. a ValueUse class (206, Fig. 2) that has a many-to-one association with the base class in the data hierarchy (Value, 207, Fig. 2) and a many-to-one association with the Case class, such that in a given simulation Case, each object of the Value class directly references one or more objects of the ValueUse class, each referenced object of the ValueUse class relates the usage of the Value object to the Cases or simulation realizations in which it is used, including the given Case.